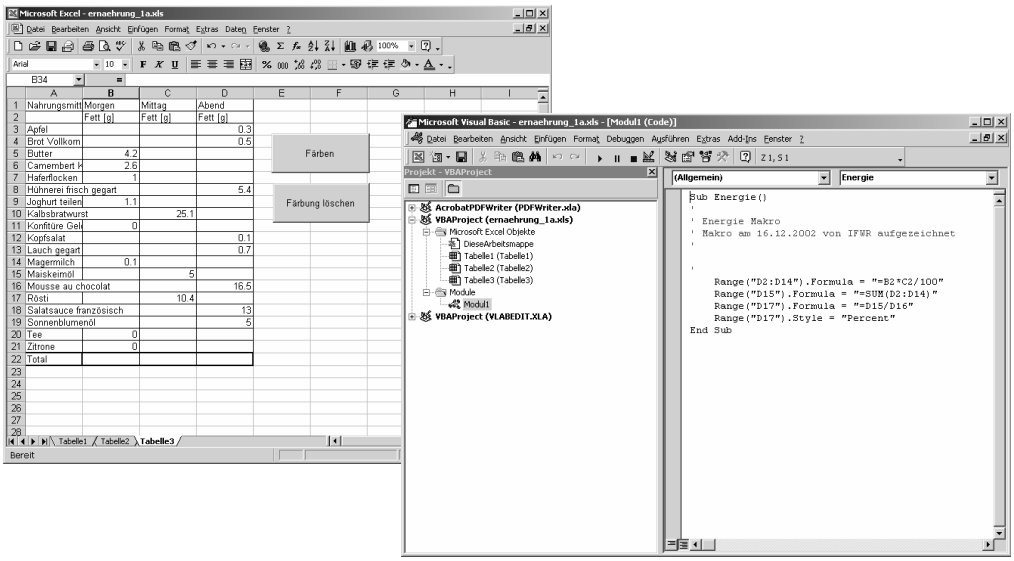
Makroprogrammierung mit MS Excel

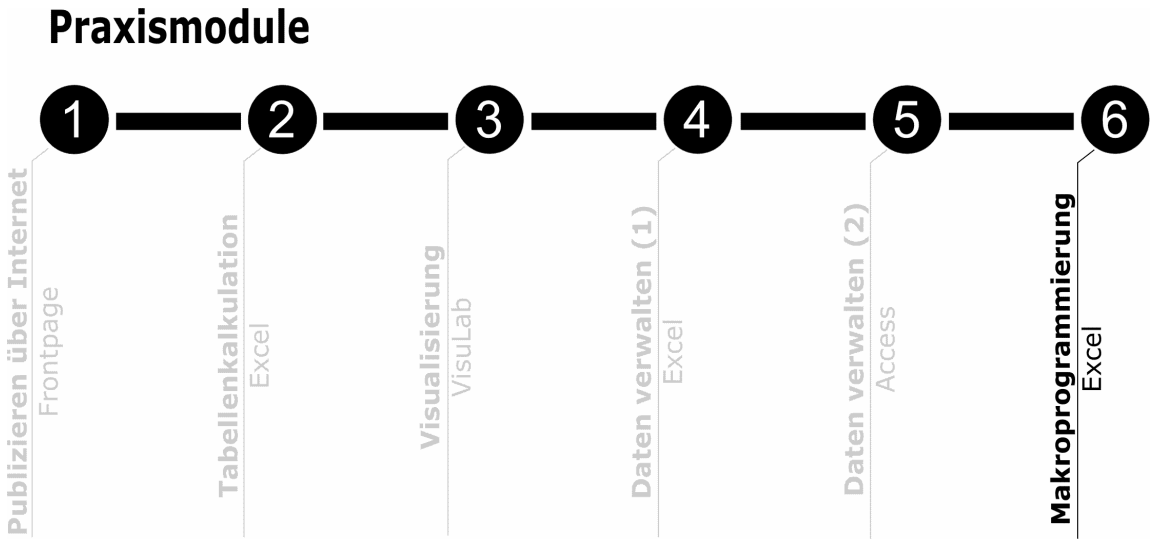
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Institut für Computational Science
Prof. Dr. H. Hinterberger

evim ETHz.ch

Praxismodul 6

Makroprogrammierung mit MS Excel





Internetadresse: **www.evim.ethz.ch**

Wie bearbeite ich dieses Modul?

Dieses Praxismodul bearbeiten Sie am effizientesten, wenn Sie die folgenden drei Teile in angegebener Reihenfolge angehen:

Begriffe:

In diesem Praxismodul werden folgende Begriffe behandelt:

Makro	**Algorithmus**	**Visual Basic**
Prozedur	**Fallunterscheidung**	**VBA**
Compiler	**Schleife**	
Objektorientierung	**Programmiersprache**	
Syntax	**Programm**	

Teil A: Einführung

Wozu Makros in Excel einsetzen?

Obwohl Excel ein sehr leistungsfähiges Programm ist, werden Sie als fortgeschrittene Anwender für spezielle Probleme massgeschneiderte Lösungen benötigen, die mit den Standardbefehlen von Excel nicht oder nur sehr aufwändig realisierbar sind. Wenn Sie beispielsweise immer gleich aufgebaute Tabellen mit neuen, aktuellen Daten auswerten müssen, kann ein Programm diese Routinearbeit für Sie erledigen. Aus diesen Bedürfnissen heraus ist schon in frühen Versionen von Excel ein Werkzeug namens **Makro** entstanden, welches das Aufzeichnen und Abspielen von allen Aufgaben ermöglicht, die Sie mit Excel erledigen.

Die Bezeichnung **Makro** stammt aus dem Gebiet der Programmierung und bezeichnet eine relativ kurze Folge von Anweisungen einer gegebenen Programmiersprache. Diese Folge wird unter einem eindeutigen Namen gespeichert und jedes Mal, wenn in einem Programm dieser Name aufgerufen wird, läuft diese Anweisungsfolge ab (**Abb. 1**).

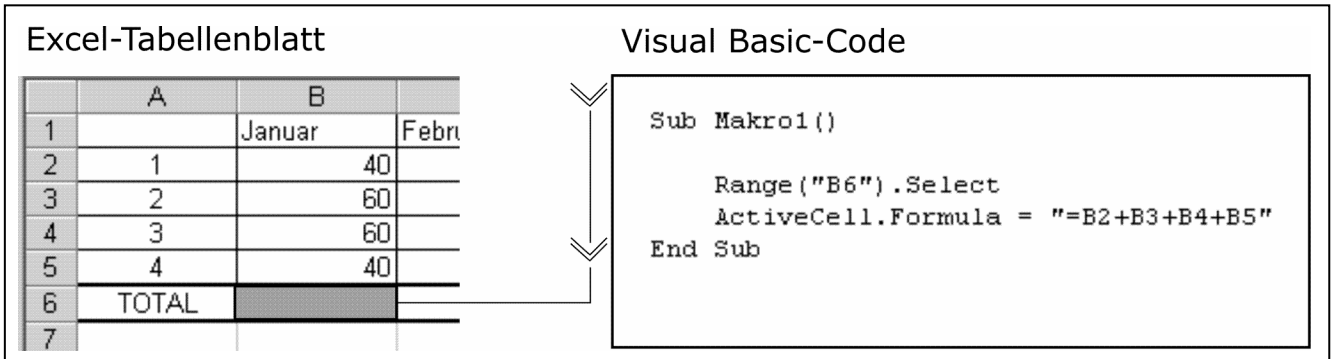
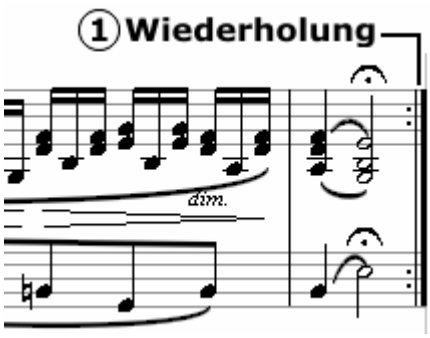


Abb. 1: Ein Makro zur Berechnung einer Summe in Zelle B6. Das Makro wird als Anweisungsfolge in der Programmiersprache *Visual Basic* aufgezeichnet und kann zusätzlich „von Hand“ mit weiteren Befehlen ergänzt werden.

Alle Aktionen die Excel ausführt, sind schlussendlich Anweisungen einer Programmiersprache, das heisst, wenn solche Aktionen aufgezeichnet werden, dann werden im Hintergrund kleine Programmsequenzen, eben Makros, niedergeschrieben und gespeichert. In Excel wird dazu die Programmiersprache **Visual Basic für Anwendungen (VBA)** verwendet. Sie stellt eine Kombination aus Excel-spezifischen Elementen (z.B. Zellen, Tabellenkalkulations-Funktionen etc.) und der Programmiersprache Visual Basic dar. VBA ist in sämtlichen heutigen Office-Anwendungen einsetzbar, also auch in Word, Access und PowerPoint.

Was ist ein Programm?

In diesem Grundmodul versuchen wir, Ihnen einige Eindrücke davon zu vermitteln, was ein Programm ist. Sie werden sehen, dass ein Programm nichts anderes ist, als eine Folge von Anweisungen, die der Reihe nach ausgeführt werden. Ein Programm hat starke Analogien zu Anweisungen, die an Personen erteilt werden, wie z.B. Rezepte oder Partituren. Man beginnt mit der ersten Anweisung und befolgt eine nach der anderen, bis die Sequenz abgeschlossen ist. Oft kommt es vor, dass solche Sequenzen oder Teile davon übersprungen oder sogar wiederholt ausgeführt werden (**Abb. 2**).



REZEPT FUER EINE LASAGNE

- 1 Schalotte und 1 Knoblauchzehe fein hacken. Wenn Sie keine Schalotten haben, dann nehmen Sie Zwiebeln. ②
- Butter in einem Topf heiss werden lassen, 100g Hackfleisch und 3 EL Tomatenmark zufügen und gut verrühren. Mit Salz, Pfeffer und Paprika würzen und das Ragout 20 Minuten kochen lassen.
- Bereiten Sie eine Sauce-Béchamel vor (Siehe S. 32). ③
- In eine gebutterte Auflaufform abwechselnd Lasagne, Ragout und Béchamelsauce schichten.
- Im vorgeheizten Backofen bei 180°C ca. 30 Minuten backen.

Abb. 2: Charakteristik von Programmen anhand der Analogie zu Anweisungen an Personen: *Wiederholung, Bedingungen prüfen und Methoden abrufen*. In der musikalischen Notation können, wie beim Programmieren auch, Teile der Anweisungskette wiederholt werden ①. Bei einem Rezept können Bedingungen geprüft werden, wie z.B. „Wenn Sie keine Schalotten haben, dann nehmen Sie Zwiebeln" ②. Es kann auch auf ein anderes Rezept (Methode) verwiesen werden, das ein Bestandteil Ihres Rezeptes ist, wie z.B. „ bereiten Sie eine Sauce Béchamel vor, siehe Seite 32" ③.

Bei der Computer-Programmierung werden die Anweisungen an eine Maschine erteilt und automatisch ausgeführt. Zusammenhängende Befehlsfolgen können mit einem eindeutigen Bezeichner benannt werden, damit durch Aufrufen dieses Namens die entsprechende Befehlsfolge an beliebiger Stelle in einem Programm zur Ausführung gebracht werden kann. So gruppierte Befehlsfolgen werden **Prozedur** genannt. Die Anweisungen werden in einer speziellen Sprache geschrieben, der **Programmiersprache**. *Visual Basic* ist eine von vielen solcher Sprachen, die zu den anwendungsnahen Sprachen gezählt werden, da sie sich sehr nahe an einer Anwendung (z.B. Excel oder Access) orientiert. Programmiersprachen haben strikte Regeln, die sogenannte **Syntax**, die den Aufbau der Sätze oder Wörter, die zur Sprache gehören, definieren. Ein **Compiler**, der auf der Syntax einer Programmiersprache basiert, übersetzt ein vollständiges Programm in eine Maschinensprache, die letztlich aus einer Folge von Nullen und Einsen besteht und für Instruktionen codiert, welche der Computer ausführen kann.

Was heisst objektorientiert Programmieren?

Mit Hilfe der objektorientierten Programmierung können verschiedene Probleme der realen Welt als Modell besser oder einfacher abgebildet werden, als mit nicht-objektorientierten Programmiersprachen (wie z.B. C oder Pascal). Ein Textverarbeitungsprogramm modelliert beispielsweise Wörter, Seiten, Zeilen und Überschriften, ein Buchungssystem für Fluglinien Flüge, Sitzplätze, Zeiten und Ziele.
In der Tabellenkalkulation entspricht ein Objekt einer Auswahl oder einer Gruppierung von Zellen zusammen mit Anweisungen, die auf diese Zellen angewendet werden können. Das heisst, dass wir in einer Zelle oder einem Zellbereich eine Funktion (z.B. Formel) anwenden, welche als Objekt-Eigenschaft angegeben wird (**Abb. 3**).

Abb. 3: Objektorientiertes Programmieren in VBA. Objekt und Eigenschaft werden durch einen Punkt voneinander getrennt dargestellt. Das Objekt "Zelle B3" wird durch die Eigenschaft "Formel" manipuliert, indem dieser Eigenschaft der Wert "=B1+B2" zugewiesen wird.

Was gehört zu diesem Praxismodul?

Dieses Modul behandelt eine Einführung in die Automatisierung von Befehlen der Anwendungssoftware Excel:

- Makro-Aufzeichnungsfunktion
- Eingabe von Hand mit dem Code-Editor
- Einsatz von Schaltflächen und Meldungsfenstern

Ausserdem erklärt es einige Grundbegriffe der objektorientierten Programmierung:

- Objekte und deren Eigenschaften
- Fallunterscheidung mit if-Bedingungen
- Wiederholungen mit Hilfe von Schleifen

Was gehört nicht dazu?

Das Erlernen einer Programmiersprache erfordert viel Übung. Erwarten Sie deshalb nicht, dass Sie nach diesem Praxismodul eine Programmiersprache vollständig beherrschen. Der Einsatz einer anwendungsnahen Programmiersprache hat den Vorteil, dass Sie relativ schnell mit einem Minimum an Vorwissen praktisch loslegen können. Ferner möchten wir, dass auch jene unter Ihnen etwas profitieren können, die nicht tiefer ins Programmieren einsteigen möchten.

Teil B: *E.Tutorial*

Arbeiten Sie das *E.Tutorial* Praxis 6 durch!

Sie finden das *E.Tutorial* auf Ihrer **CD-ROM** oder über **http://www.evim.ethz.ch**.

Im *E.Tutorial* Praxis 6 lernen Sie...

- mit Hilfe der **Makro-Aufzeichnungsfunktion** von Excel sich wiederholende Berechnungsfunktionen automatisieren (Lektionen 1 und 2).
- Makrocode im **Code-Editor** von Hand bearbeiten und erste Elemente einer objektorientierten Programmiersprache kennen (Lektion 3 und 4).
- ein erstes Programmelement umsetzen: eine **Fallunterscheidung** mit "Wenn...dann..." (Lektionen 5).
- ein zweites Programmelement umsetzen: eine **Schleife** (Lektion 6).
- **Schaltflächen** erstellen, mit der Programme ausgelöst werden können (Lektion 7).
- das Einblenden von **Meldungsfenstern** programmieren, um einen Anwender zu informieren (Lektion 8).

System-Voraussetzungen

Computer mit MS Excel Version 97 oder höher. Für das Herunterladen von Beispieldateien brauchen Sie eine Internetverbindung.
Weitere Hinweise finden Sie auf dem Blatt "Informationen zum Aufbau der Praxismodule"!

Vorsicht: Viren-Schutz

Da sich viele Viren ebenfalls der Makrofunktion bedienen, wurden in die Office-Anwendungen verschiedene Sicherheitsüberwachungen integriert. Falls Sie ein Makro nicht ausführen können, liegt es wahrscheinlich daran, dass unter Extras > Makro > Sicherheit die **Makrosicherheit** auf **Hoch** eingestellt ist. Setzen Sie die Sicherheit deshalb für dieses Praxismodul auf **Mittel**, dann werden Sie vor dem Öffnen "makrohaltiger" Dokumente gefragt, ob Sie deren Ausführung zulassen wollen

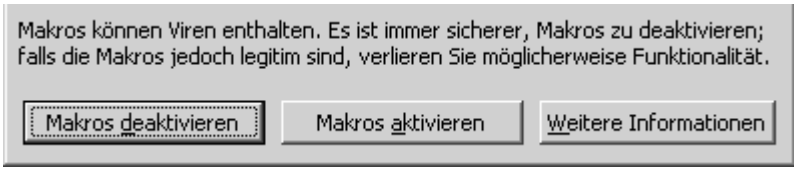


Abb. 4: Dieses Fenster erscheint, wenn Sie die Makrosicherheit auf "Mittel" setzten. Somit können Sie wählen, ob Sie ein Makro aktivieren wollen oder nicht.

Teil C: Testaufgabe

1. Einführung

Sie haben in Praxis 3 (Visualisierung) mit Hilfe von Permutationsmatrizen mehrdimensionale Daten visualisiert. Nach dem Laden einer Datensammlung in Excel erhielten Sie in VisuLab in jeder Dimension ein Säulendiagramm (Abbildung 5).

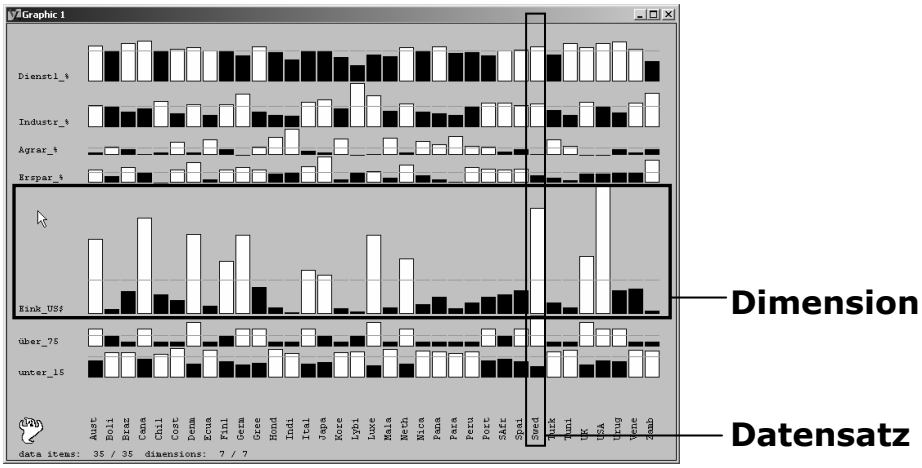


Abb. 5: Permutationsmatrix. Für jede der 7 Dimensionen wird ein Säulendiagramm erstellt.

Vielleicht waren Sie sich damals noch nicht bewusst, dass Sie auch bei diesem Programm nach dem Wählen des Menüpunktes "Permutationsmatrix" im Hintergrund eine Befehlsfolge ausgelöst haben, die in einer Programmiersprache geschrieben ist.

Welche Anweisungen könnten hinter einer solchen Permutationsmatrix stecken?

Ausgangspunkt sollen Daten einer Tabellenkalkulation sein. Für die Daten jeder Dimension (Spalten) werden folgende Anweisungen durchgeführt:

1. Berechnung des Mittelwertes
2. Für jeden Wert entscheiden, ob er grösser oder kleiner ist als der Mittelwert
3. Nach der Grösse sortieren
4. Erstellen eines Säulendiagramms

2. Vorbereitendes zur Testaufgabe

- Laden Sie die Datei **socioeco_6.xls** von unserer Homepage auf Ihren Rechner.
- Setzen Sie, falls Sie das noch nicht getan haben, in Excel die Makrosicherheit auf **Mittel** (Details siehe Teil B).

3. Aufgaben

Sie haben im *E.Tutorial* eine Einführung in die Automatisierung von Befehlen der Anwendungssoftware Excel erhalten und einige Grundbegriffe der objektorientierten Programmierung kennen gelernt. Ihre Aufgabe wird es nun sein, mit Hilfe der im *E.Tutorial* angeeigneten Fähigkeiten, Makros zu programmieren, welche in Excel per Mausklick auf eine Schaltfläche folgende Funktionen liefern (siehe Abbildung 6):

- einen Mittelwertvergleich (1)
- löschen des Mittelwertvergleichs (2)
- ein sortiertes Säulendiagramm (eine Zeile der Permutationsmatrix) (3)

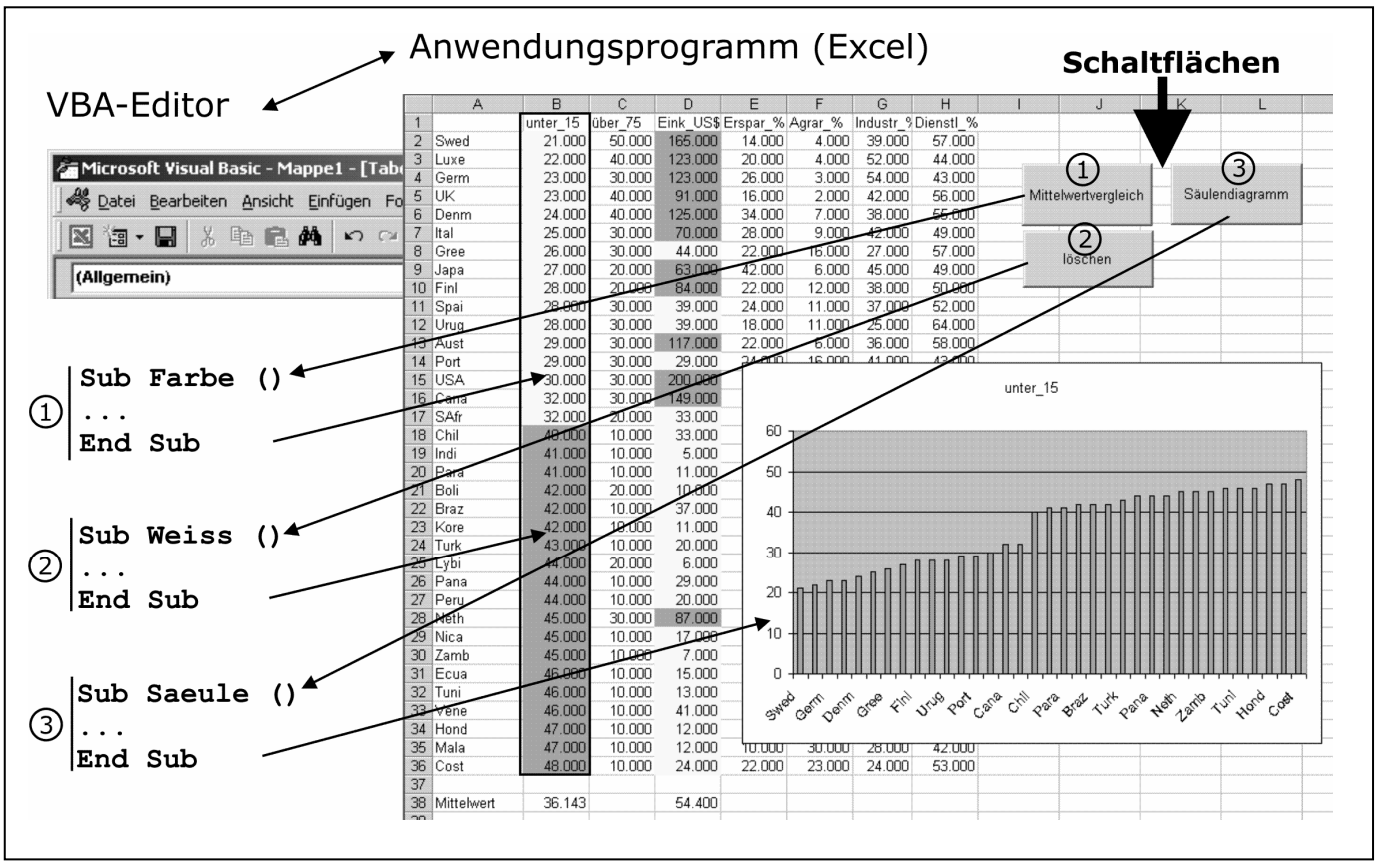


Abb. 6: Makroprogrammierung im VBA-Editor von Excel. Über Schaltflächen im Anwendungsprogramm werden die Makros aufgerufen. Etwa so sollte Ihre Testaufgabe schlussendlich aussehen.

Vorgehen

Die Testaufgabe besteht aus 2 Teilen (gemäss der unter 1. festgelegten Programmierschritte):

Teil A: Mittelwertvergleich. Jeder Wert einer Spalte (Dimension) soll mit dem Dimensions-Mittelwert verglichen werden, um deren Zellen in zwei unterschiedlichen Farben zu visualisieren.

Teil B: Sortiertes Säulendiagramm erstellen. Die Daten der ersten Spalte werden sortiert und in einem Säulendiagramm dargestellt.

Hinweis: Beachten Sie die FAQ's zur Testaufgabe unter 5.!

Teil A: Mittelwertvergleich in einer Dimension

Erstellen Sie im **Visual Basic Editor** von Excel ein Makro mit folgenden Eigenschaften:

- Berechnung des Mittelwertes der Daten einer Dimension (z.B. unter_15).
- Färbung der Zellen abhängig vom Mittelwert (z.B. rot für > Mittelwert und gelb für < Mittelwert).
- eine Schaltfläche (Command-Button), welche das Makro auslöst (siehe Abb. 6)

Erstellen Sie ein **zweites Makro** inklusive Schaltfläche, welches die Färbung der Zellen des ersten Makros rückgängig macht (resp. wieder weiss einfärbt).

Fakultativ: Erstellen Sie Mittelwertsvergleiche von weiteren Dimensionen.

Teil B: Sortiertes Säulendiagramm erstellen

Erstellen Sie mit Hilfe der **Makro-Aufzeichnungsfunktion** von Excel ein Makro mit folgenden Eigenschaften:

- Sortieren der Daten einer Dimension (z.B. unter_15).
- Erstellen eines Säulendiagramms dieser Daten.
- eine Schaltfläche (Command-Button), welche das Makro auslöst (siehe Abb. 6)

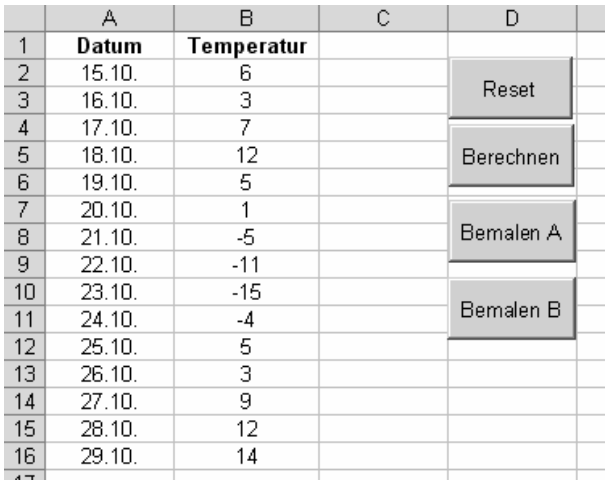
Integrieren Sie ein **Meldungsfenster** (Message-Box), welches den Anwender darüber informiert, was das Makro bewirkt.

4. Form und Bedingungen

Führen Sie einer Assistentin oder einem Assistenten Ihre funktionierenden Makros vor und kommentieren Sie die einzelnen Programmierschritte im Visual Basic-Editor.

- Welche Programmierelemente benötigen Sie für das Einfärben der Zellen abhängig vom Mittelwert?

- Wie werden Schaltflächen (Command-Buttons) programmiert?
- Was passiert nach dem Anklicken folgender Schaltflächen?



	A	B	C	D
1	**Datum**	**Temperatur**		
2	15.10.	6		Reset
3	16.10.	3		
4	17.10.	7		Berechnen
5	18.10.	12		
6	19.10.	5		Bemalen A
7	20.10.	1		
8	21.10.	-5		Bemalen B
9	22.10.	-11		
10	23.10.	-15		
11	24.10.	-4		
12	25.10.	5		
13	26.10.	3		
14	27.10.	9		
15	28.10.	12		
16	29.10.	14		

Diese Excel-Datei temperatur.xls kann auf der Homepage heruntergeladen und ausprobiert werden.

```
Private Sub Reset_Button_Click()
    Reset
End Sub
Private Sub Berechnen_Button_Click()
    Range("C4:C16").Formula = "=average(b2:b4)"
End Sub
Private Sub BemalenA_Button_Click()
    For Each Zelle In Range("B2:B16")
        Zelle.Interior.ColorIndex = Farbe(Zelle.Value)
    Next
End Sub
Private Sub BemalenB_Button_Click()
    For Each Zelle In Range("C4:C16")
        Zelle.Interior.ColorIndex = Farbe(Zelle.Value)
    Next
End Sub

Sub Reset()
    For Each Zelle In Range("B2:B16")
        Zelle.Interior.ColorIndex = 2
    Next
    For Each Zelle In Range("C2:C16")
        Zelle.Interior.ColorIndex = 2
        Zelle.Value = ""
    Next
End Sub
Function Farbe(Wert)
    Farbe = 23
    If Wert >= -10 Then Farbe = 28
    If Wert >= -5 Then Farbe = 33
    If Wert >= 0 Then Farbe = 37
    If Wert >= 5 Then Farbe = 36
    If Wert >= 10 Then Farbe = 44
    If Wert >= 15 Then Farbe = 45
    If Wert >= 20 Then Farbe = 46
    If Wert >= 25 Then Farbe = 3
    If Wert >= 30 Then Farbe = 7
End Function
```

Die Begriffe dieses Praxismoduls sollten Sie mit einfachen Worten erklären können.

Füllen Sie die **online-Kursevaluation** aus und senden Sie das Formular an uns zurück. Sie finden den Link am Ende der Lektion 8 des *E.Tutorials* oder direkt unter:

http://www.evim.ethz.ch/evaluation/e6/survey_praxis_6.html

5. FAQ's zur Testaufgabe

- *Wie stelle ich die Spaltenbezeichnung von Zahlen in Buchstaben um (von Z1S1 zu A1)?*
  Extras > Optionen...> Allgemein. Das Kästchen bei Z1S1-Bezugsart abwählen.
- *Wie lautet die Funktion zur Berechnung des Mittelwerts?*
  Die Funktion zur Berechung des Mittelwerts im Bereich B1:B12 lautet: "=AVERAGE(B1:B12)"

6. Literatur

Prudenzi, P.S. VBA mit Excel 2000 lernen. Einstieg in die Welt der Makroprogrammierung. Addison-Wesley (2000)

Schels, I. Big Hag - Programmieren lernen mit Visual Basic. Markt und Technik (2001)